CS101. A15 - Inventory Management

In this exercise, you will be managing an inventory system using Dictionaries in Python. The inventory should be organized by the item name and it should keep track of the number of items available.

You will have to handle adding items to an inventory. Each time an item appears in a given list, increase the item's quantity by 1 in the inventory. Then, you will have to handle deleting items from an inventory.

To finish, you will have to implement a function which returns all the key-value pairs in an inventory as a list of tuples.

1. Create an inventory based on a list

Implement the create_inventory() function that creates an "inventory" from a list of items. It should return a dict containing each item name paired with their respective quantity.

```
>>> create_inventory(["coal", "wood", "wood", "diamond", "diamond",
"diamond"])

{"coal":1, "wood":2 "diamond":3}
```

2. Add items from a list to an existing dictionary

Implement the add_items() function that adds a list of items to an inventory:

```
>>> add_items({"coal":1}, ["wood", "iron", "coal", "wood"])

{"coal":2, "wood":2, "iron":1}
```

3. Decrement items from the inventory

Implement the decrement_items(<items>) function that takes a list of items. The function should remove one from the available count in the inventory for each time an item appears on the list:

```
>>> decrement_items({"coal":3, "diamond":1, "iron":5}, ["diamond", "coal",
"iron", "iron"])

{"coal":2, "diamond":0, "iron":3}
```

Item counts in the inventory should not fall below 0. If the number of times an item appears on the list exceeds the count available, the quantity listed for that item should remain at 0 and additional requests for removing counts should be ignored.

```
>>> decrement_items({"coal":2, "wood":1, "diamond":2}, ["coal", "coal",
"wood", "wood", "diamond"])

{"coal":0, "wood":0, "diamond":1}
```

4. Remove an item entirely from the inventory

Implement the remove_item(<inventory>, <item>) function that removes an item and its count entirely from an inventory:

```
>>> remove_item({"coal":2, "wood":1, "diamond":2}, "coal")
```

```
{"wood":1, "diamond":2}
```

If the item is not found in the inventory, the function should return the original inventory unchanged.

```
>>> remove_item({"coal":2, "wood":1, "diamond":2}, "gold")

{"coal":2, "wood":1, "diamond":2}
```

5. Return the inventory content

Implement the list_inventory() function that takes an inventory and returns a list of (item, quantity) tuples. The list should only include the available items (with a quantity greater than zero):

```
>>> list_inventory({"coal":7, "wood":11, "diamond":2, "iron":7, "silver":0})

[('coal', 7), ('diamond', 2), ('iron', 7), ('wood', 11)]
```

Code Template (dicts.py)

```
"""
Inventory management
using dictionaries
"""

def create_inventory(items):
    """

    :param items: list - list of items to create an inventory from.
    :return:  dict - the inventory dictionary.
    """
    pass

def add_items(inventory, items):
    """

    :param inventory: dict - dictionary of existing inventory.
    :param items: list - list of items to update the inventory with.
    :return:  dict - the inventory dictionary update with the new items.
    """
    pass

def decrement_items(inventory, items):
    """

    :param inventory: dict - inventory dictionary.
    :param items: list - list of items to decrement from the inventory.
    :return:  dict - updated inventory dictionary with items decremented.
    """
    pass

def remove_item(inventory, item):
    """
    :param inventory: dict - inventory dictionary.
    :param item: str - item to remove from the inventory.
    :return:  dict - updated inventory dictionary with item removed.
    """
    pass
```

```
def list_inventory(inventory):
      """

      :param inventory: dict - an inventory dictionary.
      :return: list of tuples - list of key, value pairs from the inventory
dictionary.
      """
      pass
```

Sample output

```
Shown in the description.
```

Test Cases

Function	**Inputs**	**Output**	
create_inventory	["wood", "iron", "iron", "diamond", "diamond"]	{"wood": 1, "iron": 2, "diamond": 2}	
create_inventory	["coal", "wood", "wood", "diamond", "diamond", "diamond"])	{"coal":1, "wood":2 "diamond":3}	
add_items	{ }, ["iron", "iron", "diamond"]	{"iron": 2, "diamond": 1}	
add_items	{"coal":1}, ["wood", "iron", "coal", "wood"]	{"coal":2, "wood":2, "iron":1}	
add_items	{"wood": 2, "gold": 1, "diamond": 3}, ["wood", "gold", "gold"]	{"wood": 3, "gold": 3, "diamond": 3}	
add_items	{"iron": 1, "diamond": 2}, ["iron", "wood", "wood"]	{"iron": 2, "diamond": 2, "wood": 2}	
add_items	{"wood": 4, "iron": 2}, ["iron", "iron"]	{"wood": 4, "iron": 4}	
decrement_items	{"coal":3, "diamond":1, "iron":5}, ["diamond", "coal", "iron", "iron"]	{"coal":2, "diamond":0, "iron":3}	
decrement_items	{"coal":2, "wood":1, "diamond":2}, ["coal", "coal", "wood", "wood", "diamond"]	{"coal":0, "wood":0, "diamond":1}	
decrement_items	{"iron": 3, "diamond": 4, "gold": 2}, ["iron", "iron", "diamond", "gold", "gold"]	{"iron": 1, "diamond": 3, "gold": 0}	
decrement_items	{"wood": 2, "iron": 3, "diamond": 1}, ["wood", "wood", "wood", "iron", "diamond", "diamond"]	{"wood": 0, "iron": 2, "diamond": 0}	
remove_item	{"coal":2, "wood":1, "diamond":2}, "coal"	{"wood":1, "diamond":2}	
remove_item	{"coal":2, "wood":1, "diamond":2}, "gold"	{"coal":2, "wood":1, "diamond":2}	
remove_item	{"iron": 1, "diamond": 2, "gold": 1}, "diamond"	{"iron": 1, "gold": 1}	
remove_item	{"iron": 1, "diamond": 2, "gold": 1}, "wood"	{"iron": 1, "gold": 1, "diamond": 2}	
list_inventory	{"coal":7, "wood":11, "diamond":2, "iron":7, "silver":0}	[('coal', 7), ('diamond', 2), ('iron', 7), ('wood', 11)]	

list_inventory	{"coal": 15, "diamond": 3, "wood": 67, "silver": 0}	[("coal", 15), ("diamond", 3), ("wood", 67)]	

****