# Module 6.5 Function args-kwargs: Coding Questions with Hints

## Question 1: Write a function `sum\_numbers` that sums any number of arguments passed to it.

Hint: Use \*args to accept any number of positional arguments and sum them up.

## Question 2: Create a function `greet\_all` that accepts any number of named arguments and prints a greeting for each.

Hint: Use \*\*kwargs to accept any number of keyword arguments and iterate over them to print greetings.

## Question 3: Implement a function `multiply` that accepts any number of numeric arguments and returns their product.

Hint: Utilize \*args to work with multiple numeric inputs and calculate their product.

## Question 4: Define a function `person\_info` that takes a mandatory name argument and any number of additional keyword arguments representing person details.

Hint: Start the function definition with a normal positional argument for the name, then use \*\*kwargs for additional details.

## Question 5: Write a function `concatenate\_strings` that concatenates and returns any number of string arguments passed to it.

Hint: Employ \*args to collect multiple string inputs and concatenate them.

## Question 6: Create a function `min\_max` that returns the minimum and maximum of any number of numeric arguments.

Hint: Use \*args to accept a variable number of arguments, then use the min and max functions.

## Question 7: Implement a function `construct\_dict` that constructs a dictionary from an arbitrary number of keyword arguments.

Hint: Make use of \*\*kwargs to collect keyword arguments into a dictionary and return it.

## Question 8: Define a function `search\_dict` that searches for a key in a dictionary and returns its value, or a default value if the key is not found. The function should accept the dictionary, the key to search for, and the default value as arguments.

Hint: Start with a normal parameter for the dictionary, then use \*args or \*\*kwargs for the key and the default value.

## Question 9: Write a function `merge\_dicts` that merges any number of dictionaries into a single dictionary and returns it.

Hint: Utilize \*\*kwargs to combine several dictionaries into one.

## Question 10: Create a decorator function `type\_checker` that checks if the arguments passed to another function match the types specified in the decorator arguments.

Hint: The decorator should accept the expected types as arguments and use \*args and \*\*kwargs to inspect the types of the actual function arguments.