**Practice Problem Set**

1. Function Basics

**Problem:** Write a function named **calculate\_area** that takes two arguments: **length** and **width**. The function should return the area of a rectangle. Ensure that your function works correctly by calling it with different values for length and width.

2. Default Arguments

**Problem:** Modify the **calculate\_area** function to have a default value of 1 for both **length** and **width**. Test the function with no arguments, one argument, and two arguments.

3. Variable-length Arguments (\*args)

**Problem:** Write a function **concatenate\_strings** that takes a variable number of string arguments and concatenates them into a single string with spaces between each argument. Test the function with different numbers of strings.

4. Keyword Arguments (\*\*kwargs)

**Problem:** Create a function **build\_vehicle** that accepts various keyword arguments like **make**, **model**, **year**, **color**, etc., and returns a dictionary of these vehicle attributes.

5. Class Creation

**Problem:** Define a class **Book** with attributes **title**, **author**, and **publication\_year**. Add an instance method **get\_description** that returns a string containing the title, author, and year.

6. Inheritance

**Problem:** Create a subclass of **Book** named **EBook**. Add an attribute **file\_size** and a method **is\_large** that returns **True** if the file size is greater than 100MB.

7. Using super()

**Problem:** In the **EBook** class, override the **get\_description** method to include the file size. Use **super()** to reuse the **get\_description** method of the parent **Book** class.

8. Combining Function Arguments

**Problem:** Write a function **create\_user\_profile** that takes a username, an email, a variable number of interests (using \*args), and arbitrary user details (using \*\*kwargs). The function should return a user profile dictionary containing all this information.

9. Error Handling in Functions

**Problem:** Modify the **calculate\_area** function to handle cases where the input arguments are not numbers. Use a try-except block to catch exceptions and print an error message.

10. Practical Application

**Problem:** Create a class **TemperatureConverter** with two static methods: **celsius\_to\_fahrenheit** and **fahrenheit\_to\_celsius**. Each method should take a temperature value as an argument and return the converted temperature.