Tasks for Lab 4 | DA108 Python Programming

Learning Objectives

- Demonstrate problem solving using the concepts of conditionals and loops
- Use the while and for loops of Python
- Using Python for assisting a user, interactively

Problem 1: Factorial Calculator

Using the understanding of looping, develop a Python program that calculates the factorial of a given number. The factorial of a non-negative integer n is the product of all positive integers less than or equal to n.

Problem 2: Prime Number Checker

Create a Python program that takes a number as input and checks whether it is prime or not. A prime number is a positive integer greater than 1 that is divisible only by 1 and itself.Username Validation

Problem 3: Guessing Game

Create a Python program that assigns an integer number (between 0-100) to a variable and prompts the user to guess the number. The program should provide hints if the user's guess is too high or too low (if absolute difference is > 10), high or low (if absolute difference is <=10), and close (if absolute difference is <=5) and keep track of the number of attempts made. If the user guesses correctly, it should output a message announcing the results and number of attempts.

Problem 4: Camel case to Snake case

In some programming languages, it is common to use camel case, also known as mixed case, for variable names when they consist of multiple words. In camel case, the first letter of the first word is lowercase, but the first letter of each subsequent word is uppercase. For example, a variable for a user's name might be called name, while a variable for a user's first name could be named firstName.

Contrary to this convention, Python recommends snake case, where words are separated by underscores (_) and all letters are lowercase. Using snake case, the same variables would be named name, first_name in Python.

Your task is to implement a program named camel.py that prompts the user for a variable name in camel case and outputs the corresponding name in snake case. You can assume that the user's input will be in camel case. If it violates this assumption, then you can output a friendly message to the user.

Problem 5: Shortening text messages

When texting, it's not uncommon to shorten words to save time or space, as by omitting vowels. In a file called shorten_message.py, implement a program that prompts the user for a str of text and then outputs that same text but with all vowels (A, E, I, O, and U) omitted, whether inputted in uppercase or lowercase.
