Python Programming Tasks

Task 1: Generate a List of Even Numbers

Create a program that generates a list of all the even numbers from 1 to 20

Task 2: Sum of Positive Numbers

Write a program that calculates the sum of all positive numbers in a list.

Task 3: Calculate Total Cost

Create a function called calculate_total that takes two parameters: price and quantity. The function should calculate the total cost and return it.

Task 4: Split Name Function

Create a function called split_name that takes a full name as a parameter (e.g., 'John Smith') and returns two values: the first name and the last name.

Task 5: Apply Operation Function

Define a function called apply_operation that takes an operation function as an argument and applies it to two other values.

Task 6: Create Person Dictionary

Create a function called create_person that takes arguments like first_name, last_name, age, and city and returns a dictionary representing a person's information.

Task 7: Multiplication Table

Write a program that prints the multiplication table for a number entered by the user. For example, if the user enters 7, the program should print the 7 times table from 1 to 10.

Task 8: FizzBuzz

Write a program that prints numbers from 1 to 100. For multiples of 3, print 'Fizz' instead of the number, and for multiples of 5, print 'Buzz.' For numbers that are multiples of both 3 and 5, print 'FizzBuzz.'

Task 9: Palindrome Checker

Write a program that checks if a given word or phrase is a palindrome (reads the same backward as forward). Ignore spaces and punctuation.

Additional Tasks for the Ambitious

Task 10: List Comprehension for Even Numbers

Create a program that generates a list of all the even numbers from 1 to 20 using list comprehension.

Task 11: Prime Number Checker

Create a program that generates a list of numbers from 1 to 100 using list comprehension. Then create a function check_if_prime_number(x) which checks whether x is a prime number (returns True or False). Finally, write another list comprehension, where you apply check_if_prime_number to every number in the previously created list and store the result to the new list if it is a prime number.