problem_sets

August 29, 2023

1 Python Problem Sets

1.0.1 1. Simple Mathematical Operations

This program takes two numbers as input from the user and prints their sum, difference, product and quotient.

[18]:
```
def calculate_operations(num1, num2):
    sum = num1 + num2
    difference = num1 - num2
    product = num1 * num2
    quotient = num1 / num2

    print(num1, "+", num2, "=", sum)
    print(num1, "-", num2, "=", difference)
    print(num1, "*", num2, "=", product)
    print(num1, "/", num2, "=", quotient)

num1 = float(input("Enter the first number: "))
num2 = float(input("Enter the second number: "))

calculate_operations(num1, num2)
```

```
5.0 + 8.0 = 13.0
5.0 - 8.0 = -3.0
5.0 * 8.0 = 40.0
5.0 / 8.0 = 0.625
```

1.0.2 2. Calculator

A simple calculator which can perform basic arithmetic operations. (+, -, *, /, %, **). It uses a while loop to perform an operation based on the users request.

[16]:
```
def calculator():
    print("Welcome! I'm a calculator.\nAnd these are the arithmetic operations␣
 ↪I can do.")
    print("Enter 'add' to perform addition.")
    print("Enter 'subtract' to perform subtraction.")
    print("Enter 'multiply' to perform multiplication.")
    print("Enter 'divide' to perform division.")
    print("Enter 'remaineder' to calculate the remainder.")
    print("Enter 'power' to calculate exponentiation.")
    print("Enter 'quit' to exit the program.")

    while True:
        instruction = input("Enter an arithmetic operation: ")

        if instruction == "add":
            num1 = float(input("Enter the first number: "))
            num2 = float(input("Enter the second number: "))
            result = num1 + num2
            print(num1, "+", num2, "=", result)
        elif instruction == "subtract":
            num1 = float(input("Enter the first number: "))
            num2 = float(input("Enter the second number: "))
            result = num1 - num2
            print(num1, "-", num2, "=", result)
        elif instruction == "multiply":
            num1 = float(input("Enter the first number: "))
            num2 = float(input("Enter the second number: "))
            result = num1 * num2
            print(num1, "*", num2, "=", result)
        elif instruction == "divide":
            num1 = float(input("Enter the first number: "))
            num2 = float(input("Enter the second number: "))
            result = num1 / num2
            print(num1, "/", num2, "=", result)
        elif instruction == "remainder":
            num1 = float(input("Enter the first number: "))
            num2 = float(input("Enter the second number: "))
            result = num1 % num2
            print(num1, "%", num2, "=", result)
        elif instruction == "power":
            num1 = float(input("Enter the base number: "))
            num2 = float(input("Enter the exponent: "))
            result = num1 ** num2
            print(num1, "**", num2, "=", result)
        elif instruction == "quit":
            print("Exited the calculator program successfully.")
            break
        else:
            print("Invalid operation. Please try again.")

calculator()
```

```
Welcome! I'm a calculator.
And these are the arithmetic operations I can do.
Enter 'add' to perform addition.
Enter 'subtract' to perform subtraction.
Enter 'multiply' to perform multiplication.
Enter 'divide' to perform division.
Enter 'remaineder' to calculate the remainder.
Enter 'power' to calculate exponentiation.
Enter 'quit' to exit the program.
21.0 + 44.5 = 65.5
12.0 - 30.0 = -18.0
8.0 * 9.0 = 72.0
6.0 / 4.0 = 1.5
13.0 % 5.0 = 3.0
5.0 ** 3.0 = 125.0
Invalid operation. Please try again.
Exited the calculator program successfully.
```

1.0.3 3. Multiplication Table

A program that prints a multiplication table for a given number. It uses a for loop to multiply the number inputted by the user with numbers from 1 up to 12.

[22]:

```python
def multiplication_table(num):
    print("Multiplication Table for", num)

    for i in range(1, 13):
        result = num * i
        print(num, "x", i, "=", result)

number = int(input("Enter a number: "))
multiplication_table(number)
```

```
Enter a number:  23
Multiplication Table for 23
23 x 1 = 23
23 x 2 = 46
23 x 3 = 69
23 x 4 = 92
23 x 5 = 115
23 x 6 = 138
23 x 7 = 161
23 x 8 = 184
23 x 9 = 207
23 x 10 = 230
23 x 11 = 253
23 x 12 = 276
```

1.0.4 4. Area and Perimeter of a rectangle

This program that calculates the area and perimeter of a rectangle given its length and width.

[25]:

```python
def calculate_area(length, width):
    area = length * width
    return area

def calculate_perimeter(length, width):
    perimeter = 2 * (length + width)
    return perimeter

# Get the length and width from the user
length = float(input("Enter the length of the rectangle: "))
width = float(input("Enter the width of the rectangle: "))

# Calculate area and perimeter
area = calculate_area(length, width)
perimeter = calculate_perimeter(length, width)

# Print the results
print("Area:", area)
print("Perimeter:", perimeter)
```

```
Enter the length of the rectangle:  12
Enter the width of the rectangle:  8
Area: 96.0
Perimeter: 40.0
```

1.0.5 5. Vowel Counter

A program that counts the number of vowels in a given string. It uses a for loop to iterate over the string and count the vowels.

[22]:

```python
def count_vowels(string):
    vowel_count = 0
    string = string.lower()
    vowels = {'a', 'e', 'i', 'o', 'u'}

    # Count the number of vowels in the string
    for char in string:
        if char in vowels:
            vowel_count += 1

    print("Number of vowel(s):", vowel_count)

# Prompt the user and execute the function
string = input("Enter a string: ")
print("String:", string)
```

```
count_vowels(string)
```

```
String: Introduction to programming with python
Number of vowel(s): 11
```

1.0.6 6. Prime or not

This function checks if a number is prime or not. It calculates the remainder of the number inputted divided by starting from 2 and iterarting all the way up to the number inputted subtracted by one. And if it finds a result in which the remainder is zero, then the number inputted is not a prime number, otherwise it is a prime number

[23]:

```
def is_prime(number):
    if number <= 1: # Any number less than or equal to 1 is not prime.
        return False

    for i in range(2, int(number)):
        if number % i == 0:
            return False

    return True

# Prompt the user to enter a number
number = int(input("Enter a number: "))

# Check if the number is prime
if is_prime(number):
    print(number, "is indeed a prime number.")
else:
    print(number, "is not a prime number.")
```

```
23 is indeed a prime number.
```

1.0.7 7. Factorial Calculator

A function that calculates the factorial of a number using recursion.

[36]:

```
def factorial(num):
    if num == 0:  # Factorial of 0 is 1
        return 1
    else:
        return num * factorial(num - 1)

number = int(input("Enter a number: "))
result = factorial(number)
print("Factorial:", result)
```

```
Enter a number:  5
Factorial: 120
```

1.0.8 8. Pangram Checker

A function that determines whether a given string is a pangram (contains all the letters of the alphabet). Uses for loop to iterate over the string given to determine whether it is pangram or not.

[25]:

```python
import string

def pangram_checker(text):
    letters = set()

    # Makes the program case insenstive
    text = text.lower()

    for char in text:
        if char.isalpha():
            letters.add(char)

    if letters == set(string.ascii_lowercase):
        print(input_str, "is indeed a pangram.")
    else:
        print(input_str, "is not a pangram.")

input_str = input("Enter a text:")
pangram_checker(input_str)
```

```
qwertyuioplkjhgfdsazxcvbnm is indeed a pangram.
```

1.0.9 9. Phonebook

A simple dictionary-based phonebook program which uses a while loop to execute a function based on the users request.

[31]:

```python
phonebook = {}

def instructions():
    print("Welcome to your Phonebook!")
    print("Enter 'add' to add contact")
    print("Enter 'search' to search for a contact name")
    print("Enter 'remove' to remove contact")
    print("Enter 'display' to display contacts")
    print("Enter 'quit' to exit the program")
instructions()

def add_contact(name, number):
    phonebook[name] = number
    print("Contact added successfully!")

def search_contact(name):
    if name in phonebook:
        print("Name:", name, "\nPhone number:", phonebook[name])
    else:
        print("Contact not found.")

def remove_contact(name):
    if name in phonebook:
        del phonebook[name]
        print(name + "'s Contact has been removed successfully!")
    else:
        print("Contact not found.")

def display_contacts():
    if phonebook:
        print("Phonebook contacts:")
        for name, number in phonebook.items():
            print(name + ": ", number)
    else:
        print("Phonebook is empty.")


while True:
    instruction = input("What do you want to do: ").lower()

    if instruction == 'add':
        add_contact(input("Name: ").capitalize(), int(input("Number: ")))
    elif instruction == 'search':
        search_contact(input("Name: ").capitalize())
    elif instruction == 'remove':
        remove_contact(input("Name: ").capitalize())
    elif instruction == 'display':
        display_contacts()
    elif instruction == 'quit':
        print("Exited the phonebook program successfully.")
        break
    else:
        print("Invalid entry. Please try again.")
```

```
Welcome to your Phonebook!
Enter 'add' to add contact
Enter 'search' to search for a contact name
Enter 'remove' to remove contact
Enter 'display' to display contacts
Enter 'quit' to exit the program
Contact added successfully!
Contact added successfully!
Contact added successfully!
```

```
Phonebook contacts:
Abel:  912345678
Adonay:  912344321
Biniam:  912211221
Name: Adonay
Phone number: 912344321
Biniam's Contact has been removed successfully!
Phonebook contacts:
Abel:  912345678
Adonay:  912344321
Invalid entry. Please try again.
Exited the phonebook program successfully.
```

1.0.10 10. Guess Game

A program that generates a random number between 1 and 20, and the user has to guess it. It also compares the randomly generated number with the users guess in order to give the user a hint (for instance “Too low! Try again”)

[32]:

```python
import random

def guess_number():
    target_number = random.randint(1, 20)
    attempts = 0

    while True:
        user_guess = int(input("Guess a number between 1 and 20: "))
        attempts += 1

        if user_guess == target_number:
            print("Congratulations! You guessed the number correctly.")
            print("Number of attempts:", attempts)
            break
        elif user_guess < target_number:
            print("Too low! Try again.")
        else:
            print("Too high! Try again.")

# Run the game
print("Welcome to the Number Guessing Game!")
guess_number()
```

```
Welcome to the Number Guessing Game!
Too high! Try again.
Too low! Try again.
Too low! Try again.
Too low! Try again.
Too low! Try again.
```

```
Congratulations! You guessed the number correctly.
Number of attempts: 6
```

1.1 The End