Homework Assignment: Data Types and Operators

1. **Operations with Integers:**
Write a Python program that takes two integers as input and performs the following operations:
- Addition: Add the two integers and display the result.
- Subtraction: Subtract the second integer from the first integer and display the result.
- Multiplication: Multiply the two integers and display the result.
- Division: Divide the first integer by the second integer (floating-point division) and display the result.
- Modulus: Calculate the remainder of dividing the first integer by the second integer and display the result.

2. **Rectangle Area and Perimeter:**
Write a program that calculates the area and perimeter of a rectangle. Follow these steps:
- Prompt the user to enter the length of the rectangle.
- Prompt the user to enter the width of the rectangle.
- Calculate the area by multiplying the length and width.
- Calculate the perimeter by adding the lengths of all four sides.
- Display the calculated area and perimeter.

3. **Counting Vowels and Consonants:**
Write a Python program that takes a sentence as input and counts the number of vowels and consonants. Follow these steps:
- Prompt the user to enter a sentence.
- Initialize variables to count the number of vowels and consonants.
- Iterate through each character in the sentence.
- Check if the character is a vowel or consonant using appropriate conditions.
- Increment the respective counter based on the character type.
- Display the final counts of vowels and consonants.

4. **Average of Numbers:**
Write a program that takes a list of numbers as input and calculates the average of all the numbers. Follow these steps:
- Prompt the user to enter a list of numbers separated by spaces.
- Split the input string into a list of individual numbers.
- Convert each number in the list to a float data type.
- Calculate the sum of all the numbers using the `sum()` function.
- Divide the sum by the total count of numbers to calculate the average.
- Display the calculated average.

5. **Celsius to Fahrenheit Conversion:**

Write a program that converts a temperature in Celsius to Fahrenheit. Follow these steps:

- Prompt the user to enter a temperature in Celsius.
- Convert the input to a float data type.
- Use the conversion formula: Fahrenheit = (Celsius * 9/5) + 32.
- Calculate the corresponding temperature in Fahrenheit.
- Display the converted temperature in Fahrenheit.

Examples: Control Flow with Decisions and Loops

1. **Even or Odd Numbers:**

Write a program that takes an integer as input and determines whether it is even or odd. Follow these steps:

- Prompt the user to enter an integer.
- Convert the input to an integer data type.
- Check if the number is divisible by 2 using the modulus operator (%).
- If the modulus is zero, display a message stating that the number is even.
- Otherwise, display a message stating that the number is odd.

2. **Print Numbers in a Range:**

Write a program that prints all the numbers from 1 to a given positive integer. Follow these steps:

- Prompt the user to enter a positive integer.
- Convert the input to an integer data type.
- Use a for loop to iterate from 1 to the given integer (inclusive).
- Print each number in the loop.

3. **Factorial Calculation:**

Write a program that calculates the factorial of a given positive integer. Follow these steps:

- Prompt the user to enter a positive integer.
- Convert the input to an integer data type.
- Initialize a variable to hold the factorial value, starting with 1.
- Use a for loop to iterate from 1 to the given integer (inclusive).
- Multiply the factorial variable by each iteration value.
- After the loop, display the final factorial value.

4. **Guessing Game:**

Write a program that generates a random number and asks the user to guess it. Follow these steps:

- Import the `random` module for generating random numbers.
- Generate a random number between 1 and 10 using the `random.randint()` function.
- Prompt the user to enter a guess between 1 and 10.
- Convert the input to an integer data type.

- Compare the user's guess with the random number.
- Display a message indicating whether the guess was correct, too low, or too high.
- Repeat the process until the user guesses the correct number.

5. **Sum of Numbers:**

Write a program that calculates the sum of all numbers from 1 to a given positive integer. Follow these steps:

- Prompt the user to enter a positive integer.
- Convert the input to an integer data type.
- Initialize a variable to hold the sum, starting with 0.
- Use a for loop to iterate from 1 to the given integer (inclusive).
- Add each iteration value to the sum variable.
- After the loop, display the final sum value.

Examples: Input/Output Operations

1. **Console Input and Output:**

Write a program that takes input from the user and displays output on the console. Follow these steps:

- Prompt the user to enter their name.
- Read the input using the `input()` function and store it in a variable.
- Display a welcome message that includes the user's name.
- Prompt the user to enter their age.
- Read the input and convert it to an integer data type.
- Calculate the user's birth year by subtracting their age from the current year (you can use the `datetime` module for the current year).
- Display a message that includes the user's birth year.
- Optionally, you can include additional prompts and interactions with the user.

2. **File Input and Output:**

Write a program that performs file input and output operations. Follow these steps:

- Open a file in write mode using the `open()` function and a file name (e.g., "data.txt") as a parameter.
- Prompt the user to enter a sentence or a line of text.
- Read the input using the `input()` function and store it in a variable.
- Write the input to the file using the file object's `write()` method.
- Close the file using the file object's `close()` method.
- Open the file in read mode.
- Read the content of the file using the file object's `read()` method and store it in a variable.
- Display the content on the console.
- Close the file.

- Optionally, you can perform additional file operations like appending or modifying the file content.

Examples: Code Documentation, Structure, Errors/Exceptions, Modules/Tools

1. **Code Documentation with Docstrings:**
   Write a Python function that calculates the area of a circle. Follow these steps:
   - Define a function called `calculate_area_circle` that takes the radius as a parameter.
   - Add a docstring to the function that explains its purpose and provides details about the input parameter.
   - Calculate the area of the circle using the formula `area = pi * radius^2`, where `pi` is a constant.
   - Return the calculated area.
   - Outside the function, prompt the user to enter the radius of a circle.
   - Convert the input to a float data type and call the `calculate_area_circle` function, passing the radius as an argument.
   - Display the calculated area on the console.

2. **Code Structure with Functions:**
   Write a Python program that calculates the sum of squares for a given range of numbers. Follow these steps:
   - Define a function called `sum_of_squares` that takes a start and end value as parameters.
   - Inside the function, use a loop to iterate through the range of numbers from the start to the end (inclusive).
   - Calculate the square of each number and sum them up.
   - Return the total sum of squares.
   - Outside the function, prompt the user to enter the start and end values of the range.
   - Convert the inputs to integer data types and call the `sum_of_squares` function, passing the start and end values as arguments.
   - Display the result on the console.

3. **Error Handling with Try-Except:**
   Write a program that calculates the division of two numbers. Follow these steps:
   - Prompt the user to enter two numbers: a numerator and a denominator.
   - Convert the inputs to float data types.
   - Use a try-except block to handle the possibility of a ZeroDivisionError.
   - Inside the try block, divide the numerator by the denominator and display the result.
   - Inside the except block, display a custom error message if a ZeroDivisionError occurs.

4. **Module Usage:**
   Write a program that uses the `random` module to generate a random number between 1 and 10. Follow these steps:

- Import the `random` module at the beginning of your program.
   - Use the `random.randint()` function to generate a random number between 1 and 10.
   - Display the generated random number on the console.

5. **External Module/Tool:**
   Write a program that uses the `requests` module to fetch data from a web API. Follow these steps:
   - Import the `requests` module at the beginning of your program.
   - Use the `requests.get()` function to send a GET request to a specified URL.
   - Capture the response using a variable.
   - Access the response content or retrieve specific data from the API response.
   - Process and display the retrieved data on the console.

REVIEW

Definitions/Terminology:

1. What is Python?
2. What is the difference between a list and a tuple in Python?
3. What is a dictionary in Python?
4. What is the purpose of the `if` statement in Python?
5. What is a loop in Python?
6. What is the difference between a function and a method in Python?
7. What is the purpose of the `import` statement in Python?
8. What is an exception in Python?
9. What is the difference between `==` and `is` operators in Python?
10. What is the purpose of the `len()` function in Python?

Coding Problems:

11. Write a Python program to calculate the sum of all even numbers from 1 to 100.
12. Write a Python function to check if a given string is a palindrome or not.
13. Write a Python program to find the largest number in a given list.
14. Write a Python function to calculate the factorial of a given number.
15. Write a Python program to check if a number is prime or not.
16. Write a Python function to reverse a given list.
17. Write a Python program to count the number of vowels in a given string.
18. Write a Python function to find the second smallest number in a given list.
19. Write a Python program to generate the Fibonacci sequence up to a given number of terms.
20. Write a Python function to calculate the area of a triangle given its base and height.