Tutorial 07

Guided Practice Questions

Easy Questions

1. **Write a Function to Add Three Numbers**
   - Write a function that takes three numbers as parameters and returns their sum. This will practice using multiple parameters and return statements.

2. **Create a Function to Multiply Two Numbers and Return the Result**
   - Define a function that accepts two numbers as parameters and returns their product. This simple exercise reinforces the concept of returning values from functions.

3. **Write a Function to Calculate the Average of Three Numbers**
   - Implement a function that takes three numbers as input and returns their average. This question combines addition and division with function parameters and return values.

4. **Function to Determine the Maximum of Two Numbers**
   - Write a function to find the maximum of two numbers. The function should take two numbers as parameters and return the larger number.

5. **Write a Function to Calculate the Perimeter of a Rectangle**
   - Write a function that takes the length and width of a rectangle as parameters and returns the perimeter of the rectangle.
   - Ensure you provide a brief explanation or comments within your code to explain the logic.

6. **Create a Function to Convert Fahrenheit to Celsius**
   - Define a function that accepts the temperature in Fahrenheit and returns the temperature converted to Celsius. The formula to convert Fahrenheit to Celsius is **(Fahrenheit - 32) * 5/9**.

7. **Function to Check if a Number is Prime**
   - Write a simple function to check whether a given number (greater than 1) is prime or not. The function should return **True** if the number is prime and **False** otherwise.

Unguided Practice Questions

Hard Questions

1. **Write a Function to Implement a Basic Calculator**
   - Implement a function that takes three arguments: two numbers and an operation (e.g., '+', '-', '*', '/'), and returns the result of the operation on the two numbers. Ensure your function handles division by zero gracefully.
2. **Function to Generate Fibonacci Sequence Up to n Terms**
   - Write a function that generates the Fibonacci sequence up to n terms and returns it as a list. The Fibonacci sequence is a series where the next term is the sum of the previous two terms, with the first two terms being 0 and 1, respectively.
3. **Write a Function to Implement a Simple Interest Calculator**
   - Implement a function that calculates simple interest given principal, rate, and time as parameters. The formula for simple interest is **(principal * rate * time) / 100**. This question tests the ability to work with more complex formulas.
4. **Create a Function to Calculate the Area of a Triangle Using Heron's Formula**
   - Define a function that takes the lengths of all three sides of a triangle as parameters and returns the area of the triangle. Use Heron's formula: **area = sqrt(s * (s - a) * (s - b) * (s - c))**, where **s** is the semi-perimeter of the triangle **(a + b + c) / 2**. This requires using parameters, arithmetic operations, and the **sqrt** function from the math module.
5. **Write a Function to Calculate Exponentiation Without Using the ** Operator**
   - Implement a function that takes two parameters, base and exponent, and calculates the power without using the ** operator. This will test loops or recursion in functions.
6. **Function to Check Whether a Number is Even or Odd Without Using % or / Operators**
   - Write a function to determine if a given number is even or odd, using bitwise operators instead of modulus or division. This encourages thinking about alternative ways to solve problems and the use of bitwise operations.