Week 5: Functions & Modularity

W5_Practical.cpp

Objective: Refactor code into reusable functions. **Task:** Take the "rectangle calculator" from Week 2 and refactor it. Create two functions: `calculateArea()` and `calculatePerimeter()`. Each function should accept `length` and `width` as parameters and return the calculated value. The `main` function should call these functions and print the results.

Solution:

C++

```
#include <iostream>

// Function prototype for calculateArea.
// It tells the compiler that this function exists before it's defined.
double calculateArea(double length, double width);

// Function prototype for calculatePerimeter.
double calculatePerimeter(double length, double width);

int main() {
   double length = 8.5;
   double width = 3.0;

   // Call the functions to get the results.
   double area = calculateArea(length, width);
   double perimeter = calculatePerimeter(length, width);

   std::cout << "For a rectangle with length " << length << " and width " << width << ":" << std::endl;
   std::cout << "Area: " << area << std::endl;
   std::cout << "Perimeter: " << perimeter << std::endl;

   return 0;
}

// Function definition for calculateArea.
// It takes two doubles and returns a double.
double calculateArea(double length, double width) {
   return length * width; // Return the calculated value.
}

// Function definition for calculatePerimeter.
double calculatePerimeter(double length, double width) {
   return 2 * (length + width);
}
```