Day 1: Assignment 3

Create a document that describes the design of two modular functions: one that returns the factorial of a number, and another that calculates the nth Fibonacci number. Include pseudocode and a brief explanation of how modularity in programming helps with code reuse and organization.

1. Factorial function

function factorial(n):

if n == 0 or n == 1;

return 1

else:

result = 1

for i from 2 to n;

result = result \* i

return result

1. Fibonacci function:

function Fibonacci(n):

if n <= 1:

return n

else

a, b = 0, 1

for i from 2 to n:

temp = b

b = a + b

a = temp

return b

Modularity in Programming:

Modularity in computing and programming refers to dividing a system into separate modules or components. Each module handles a specific functionality and operates independently. It simplifies design, development, testing, and maintenance by allowing you to focus on one part at a time without affecting the rest of the system. Modularity offers several benefits:

1. Code Reusability:

Code reusability refers in writing a code in such a way that it can be reused with small modification or sometimes without any modification. Modular functions can be reused in multiple parts of program.

1. Ease of maintenance:

Modular code is easier to maintain and debug because each module is self-contained and addresses a specific concern. If there is a bug or a need for enhancement, developers can focus on the specific module without affecting other parts of the program.

1. Scalability:

Modularity facilitates scalability by allowing developers to add new features or make changes to existing ones without disrupting the entire codebase. New functionalities can be implemented as separate modules and integrated seamlessly into the existing system.

1. Improved Readability:

Breaking down a program into smaller modules improves readability and comprehension. Each module can be understood independently, making the codebase more understandable for developers, which in turn enhances collaboration and teamwork.