# Objectives: Lab 01

The purpose of the first lab of BTP500 is to familiarize yourself with the Visual Studio User Interface and write a simple C++ program with functions. By the end of this lab, you should be able to:

* Successfully run 2 functions to analyze the time of the program and code of functions.

# Preface:

If you have not already done so, you will need to download the sample Lab00 file and run a simple C++ program. Get Started with Visual Studio Instructions

# LAB 01 - SUBMISSION

Explore the Function analysis in C++.

1. Write a C++ function to calculate the Modulo Operator % of 2 arguments n and d. Analyze the following function based on Week-1 lecture. Paste the code below.

int modulo(int n, int d) {  
 int factor = n / d; // 2  
 return n - (factor \* d); // 2  
}  
  
// Let n represent the value for the modulo operation  
// Let T(n) represent number of operations needed to find the result of the modulo operation  
// T(n) = 2 + 2 = 4  
// so, T(n) is O(1)  
// This means that this algorithm is in constant time.

1. Write a C++ function to find a Fibonacci series of an integer n. Analyze the following function based on Week-1 lecture. Paste the code below.

The Fibonacci Sequence is the series of numbers:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...

The next number is found by adding up the two numbers before it:

the 2 is found by adding the two numbers before it (1+1),

the 3 is found by adding the two numbers before it (1+2),

the 5 is (2+3),

and so on!

void fibonacci(int n) {  
 int previous = 0, previous2 = 1; // 2  
 std::cout << previous; // 1  
 for(int i = 0; i < n; i++) { // 1 + n + n  
 int current = previous + previous2; // 2\*n  
 previous2 = previous; // 1\*n  
 previous = current; // 1\*n  
 std::cout << ',' << current;  
 }  
}  
  
// Let n represent the number for the fibonacci series  
// Let T(n) represent number of operations needed to find the number in the fibonacci series  
// T(n) = 2 + 1 + 2n + 2n + 1n + 1n  
// T(n) = 6n + 3  
// From this we can see that the number of operations increases linearly.  
// so, T(n) is O(n)  
// This means that this algorithm is in linear time.

1. Write a C++ main program to include functions 1 and 2. List a choice to pick function 1 or function 2 and pass the values to print the result. Paste the screen shot of execution for function1 and function 2. Attach the final C++ file along with this Word file. Don’t add a .zip or RAR file.

# SUBMISSION for Lab 1 is 2 files:

1. This Word file with answers
2. The final CPP file with your name in the code header and add comment headers for functions.