

Lab 8

Course: CSE 165

All the exercises below are selected from the textbook: Thinking in C++.

1. [Exercise 5 on Page 780, Vol 1] Templatize the `fibonacci()` function on the type of value that it produces (so it can produce long, float, etc. instead of just int). [30 pts]

Introduction to Fibonacci numbers:

The formula to calculate the $(n + 1)^{\text{th}}$ number in the sequence of Fibonacci numbers can be given as $F_n = F_{n-1} + F_{n-2}$ where $n > 1$,

$F_{n-1} \rightarrow n^{\text{th}}$ Fibonacci number,
 $F_{n-2} \rightarrow (n - 1)^{\text{th}}$ Fibonacci number,
 $F_0 = 0$,
 $F_1 = 1$.

In your implementation, `fibonacci(n)` returns F_n .

Output: In `main()`, call `fibonacci(165)` to print out F_{165} .

2. [Exercise 1 on Page 397, Vol 2] Create a class (e.g., `MyClass`). Within this class, make a nested class (e.g., `MyException`) to use as an exception object. It takes a single `char*` as its argument (e.g., `MyException(const char* str) {myString = str; /* Note myString is a private data member of MyException */}`); this represents a description string. Within `MyClass`, create a member function (e.g., `myFun`) that throws this exception. And within `MyException`, create a member function (e.g., `printException`) to print out `myString`. [70 pts]

In `main()`, do the following:

- a. Create an object (e.g., `myObj`) of `MyClass`.
- b. Write a try block that calls `myObj.myFun()`.
- c. Write a catch clause that handles the exception by printing out its description string (i.e., calling the member function `printException()` of `MyException`).

Requirements:

- * Usage of spaces, blank lines, indentation, and comments for readability.
- * Descriptive names of variables, functions, structs, classes, and objects (if any).
- * Appropriate usage of structs, classes, and objects (if any).

Penalties:

- * Zero if you have possession of a copy of online solutions or work done by someone else.
- * 5-point deduction per day late