Module 7: Portfolio Milestone

Temuujin Tsogtgerel

Colorado State University Global

CSC450-1

Professor Dr. Reginald Haseltine

01/30/2025

Code:

```
#include <iostream>
#include <thread>
#include <mutex>
#include <condition_variable>

std::mutex mtx;
std::condition_variable cv;
bool firstThreadDone = false;

void countUp() {
   for (int i = 0; i <= 20; ++i) {
       std::lock_guard<std::mutex> lock(mtx);
       std::cout << "Counting Up: " << i << std::endl;
   }
   {
       std::lock_guard<std::mutex> lock(mtx);
       firstThreadDone = true;
   }
   cv.notify_one(); // Notify the second thread
}

void countDown() {
   std::unique_lock<std::mutex> lock(mtx);
   cv.wait(lock, [] { return firstThreadDone; }); // Wait until countUp()
completes

   for (int i = 20; i >= 0; --i) {
       std::cout << "Counting Down: " << i << std::endl;
   }
}

int main() {
   std::thread t1(countUp);
   std::thread t2(countDown);
```

```
    t1.join();
    t2.join();

    return 0;
}
```

Analysis Of Appropriate Concepts

Performance Issues with Concurrency

When a program runs two threads, the computer must switch between them. This switching takes time and can slow down the program. The program also uses a lock (mutex) to control access to shared data. Locks help prevent errors but can make the program slower if threads have to wait. If many threads try to run at the same time, they can slow each other down. The program also uses std::condition_variable to make sure the second thread waits for the first thread to finish. This is good for keeping order but adds extra steps, which can make the program run a little slower.

Vulnerabilities Exhibited with Use of Strings

This program does not use strings much, so it is safe from most string problems. However, if the program used char arrays instead of std::string, it could be risky. Using functions like strcpy() or sprintf() can cause buffer overflows. This means the program might write too much data into memory, which can cause crashes or security problems. A safer way to use strings in C++ is to use std::string, which helps prevent these issues.

Security of the Data Types Exhibited

The program uses int for counting. This is safe because the numbers are small. But in other programs, using int without limits can cause problems. If a number becomes too large, it can overflow and behave in unexpected ways. The program also uses bool for checking if the first thread is done. This is a simple and safe way to track the program's progress. The mutex and condition_variable help control how the threads run, making sure data is not changed at the wrong time. This prevents errors and makes the program more secure.