

Exam 1 – Problem 2

Given: Write a lambda function that for a reference to the following struct returns a 64-bit hash value of objects of that type...

Source Code:

```
//Samuel Barker
//00100768
//sbarker1@my.athens.edu
//CS 417, Exam 1, Problem 2

#include <iostream>
#include <string>
#include <functional>

struct S {
    std::string firstName;
    std::string lastName;
    std::string address;
};

// I chose the Lambda function to compute the hash value of struct S
auto hash_s = [](const S& s) {
    std::hash<std::string> hasher;
    std::size_t hashValue = 0;

    // Below, I will calculate hash values. I am applying the binary xor operator
    // to this as well.
    hashValue ^= hasher(s.firstName);
    hashValue ^= hasher(s.lastName);
    hashValue ^= hasher(s.address);

    // Cast resulting hash value to 64-bit
    return static_cast<uint64_t>(hashValue);
};

int main() {
    S myStruct{ "Samuel", "Collie", "687 Ford Street" };
    uint64_t hashResult = hash_s(myStruct);

    std::cout << "Hash value: " << hashResult << std::endl;

    return 0;
}
```