

**CS 311 Homework 1**  
**Due Monday 9/16/2019 At 11:59 PM**

Consider the following C++ program that makes use of many features that are unique to C++ and did not exist in C.

```
#include <iostream>
#include <string>
#include <cstdint>
#include <vector>

using namespace std;

enum LetterGrade {
    A = 4,
    B = 3,
    C = 2,
    D = 1,
    F = 0
};

// type T must be castable into a double
template<class T>
double getArrayAverage(vector<T>& vec) {
    double sum = 0;
    for (const auto& value : vec) {
        sum += static_cast<double>(value);
    }
    const auto avg = sum / vec.size();
    return avg;
}

void convertCharToLetterGrade(char& grade) {
    switch (grade) {
        case 'A': case 'a':
            grade = 4;
            return;
        case 'B': case 'b':
            grade = 3;
            return;
        case 'C': case 'c':
            grade = 2;
            return;
        case 'D': case 'd':
            grade = 1;
            return;
        case 'F': case 'f':
            grade = 0;
    }
}
```

```

        return;
    default:
        cout << "Warning... Invalid Character... Recording an F.\n";
        return;
    }
}

LetterGrade getLetterGradeFromAverage(const double avg) {
    if (avg >= 90)
        return LetterGrade::A;
    else if (avg >= 80)
        return LetterGrade::B;
    else if (avg >= 70)
        return LetterGrade::C;
    else if (avg >= 60)
        return LetterGrade::D;
    else
        return LetterGrade::F;
}

int main()
{
    string firstName;
    cout << "Please enter your first name: ";
    cin >> firstName;

    string lastName;
    cout << "Please enter your last name: ";
    cin >> lastName;

    int32_t numPrevCourses;
    cout << "Enter number of previous courses: ";
    cin >> numPrevCourses;
    cin.ignore();
    vector<LetterGrade> prevGrades(numPrevCourses);
    for (int32_t courseIx = 0; courseIx < numPrevCourses; ++courseIx) {
        cout << "Enter letter grade for course " << courseIx << ": ";
        char letterGrade;
        cin.get(letterGrade);
        cin.ignore();
        convertCharToLetterGrade(letterGrade);
        prevGrades.at(courseIx) = static_cast<LetterGrade>(letterGrade);
    }

    int32_t numExams;
    cout << "Enter number of exams this semester: ";
    cin >> numExams;
    cin.ignore();

```

```

vector<int32_t> examGrades(numExams);
for (int32_t examIx = 0; examIx < numExams; ++examIx) {
    cout << "Enter grade for exam " << examIx << " as an integer: ";
    cin >> examGrades.at(examIx) ;
    cin.ignore();
}

const auto fullName = firstName + " " + lastName;
cout << "Grade Report For " << fullName << ":\n";
const auto examAverage = getArrayAverage(examGrades);
cout << "Your exam average is: " << examAverage << "\n";

// get GPA with newest course added:
const auto newLetterGrade = getLetterGradeFromAverage(examAverage);
prevGrades.push_back(newLetterGrade);
const auto gpa = getArrayAverage(prevGrades);
cout << "Your latest GPA is: " << gpa << "\n";
return 0;
}

```

**Your Task:** Please rewrite this program in pure C and without any C++ elements. You may use any compiler that you would like, but your program cannot have any C++ features . You are also free to use a simple web-based C compiler such as the one here: <https://www.jdoodle.com/c-online-compiler>

Once you finish writing your program, please write a brief report (no more than a few paragraphs) describing features in the above program that are in C++ and not in C and the different work-arounds you had to come up with in order to achieve the same functionality. Please feel free to elaborate on the aspects of the C program that were difficult to implement.

Please submit your program (as a .c file) as well as your report (any text format will suffice) on BlackBoard. **Please do not email me your solutions.**