src/Student.cpp

```cpp
// Title:  Student.cpp
// Desc:   File where Student class's functions and methods are implemented
// Name:   An Tran

#include <iostream>
#include <iomanip>
#include <ios>
#include <sstream>
#include "Student.h"

// constuctors
// default
Student::Student() : sid(0), phoneNumber(0), count(0) {}

// parameterized
Student::Student(int sid, std::string fName, std::string lName, std::string address, long phoneNumber, std::string grades, int count)
    : sid(sid), fName(fName), lName(lName), address(address), phoneNumber(phoneNumber), grades(grades), count(count) {}

// getter member functions
int Student::getSID() {
    return sid;
}

std::string Student::getFName() {
    return fName;
}

std::string Student::getLName() {
    return lName;
}

std::string Student::getAddress() {
    return address;
}

long Student::getPhone() {
    return phoneNumber;
}

int Student::getCount() {
    return count;
}

int Student::getGrade(int location)
{ //return the string value of the grade at the given location
    std::istringstream iss(grades);
    std::string value;
    int counter(0);
    while (iss >> value)
    {
        if (counter++ == location)
        {
```

```
53          return stoi(value);
54          }
55      }
56      return -1;
57  }
58
59  // setter member functions
60  void Student::setSID(int newSID) {
61      sid = newSID;
62  }
63
64  void Student::setFName(std::string newFName) {
65      fName = newFName;
66  }
67
68  void Student::setLName(std::string newLName) {
69      lName = newLName;
70  }
71
72  void Student::setAddress(std::string newAddress) {
73      address = newAddress;
74  }
75
76  void Student::setPhone(long newPhoneNumber) {
77      phoneNumber = newPhoneNumber;
78  }
79
80  void Student::setCount()
81  { //set the counter to the number of grades in the string
82
83      std::istringstream iss(grades);
84      std::string value;
85
86      int counter = 0;
87      while (iss >> value)
88      {
89          counter++;
90      }
91      count = counter;
92  }
93
94  // Other functions
95
96  void Student::setStudent(int newSID, std::string newFName, std::string newLName,
    std::string newAddress, long newPhoneNumber, std::string newGrades) {
97      sid = newSID;
98      fName = newFName;
99      lName = newLName;
100     address = newAddress;
101     phoneNumber = newPhoneNumber;
102     grades = newGrades;
103 }
104
105 void Student::displayStudent() {
106     std::cout << std::string(34, '*') << std::endl;
107     std::cout << std::left << std::setw(13) << "Student ID   :" << std::right <<
    std::setw(20) << sid << std::endl;
```

```
108     std::cout << std::left << std::setw(13) << "First Name   :" << std::right << std::setw(20) << fName << std::endl;
109     std::cout << std::left << std::setw(13) << "Last Name    :" << std::right << std::setw(20) << lName << std::endl;
110     std::cout << std::left << std::setw(13) << "Address      :" << std::right << std::setw(20) << address << std::endl;
111     std::cout << std::left << std::setw(13) << "Phone Number :" << std::right << std::setw(20) << phoneNumber << std::endl;
112     std::cout << std::string(34, '*') << std::endl;
113 }
114 
115 void Student::addGrade(int grade) {
116     if (grade < 0 || grade > 100){
117         return;
118     };
119     if (!grades.empty()) {
120         grades += " ";
121     };
122     grades += std::to_string(grade);
123     setCount();
124 }
125 
126 std::string Student::convertLetterGrade(int grade) {
127     switch (grade / 10) {
128         case 10: // For 100
129         case 9:  // For 90-99
130             if (grade >= 94) return "A";
131             else return "A-";
132         case 8:  // For 80-89
133             if (grade >= 87) return "B+";
134             else if (grade >= 84) return "B";
135             else return "B-";
136         case 7:  // For 70-79
137             if (grade >= 77) return "C+";
138             else if (grade >= 74) return "C";
139             else return "C-";
140         case 6:  // For 60-69
141             if (grade >= 67) return "D+";
142             else if (grade >= 64) return "D";
143             else if (grade >= 61) return "D-";
144             else return "F"; // for 60 cause autograder broken?
145         default: // For anything below 60
146             return "F";
147     }
148 }
149 
150 std::string Student::currentLetterGrade() {
151     double sum = 0;
152     int count = 0;
153     std::istringstream iss(grades);
154     int tempGrade;
155     while (iss >> tempGrade) {
156         sum += tempGrade;
157         ++count;
158     }
159     if (count == 0) return "N/A"; // Handle no grades case
160     int average = static_cast<int>(sum / count);
```

```
161     return convertLetterGrade(average);
162 }
163 
164 void Student::listGrades() {
165     std::cout << std::string(14, '*') << std::endl;
166     std::cout << std::left << std::setw(3) << "GRD  " << std::right << std::setw(9) <
    < "Cum.Avg." << std::endl;
167     std::cout << std::string(14, '*') << std::endl;
168 
169     std::istringstream iss(grades);
170     int grade;
171     double sum = 0;
172 
173     for (int i = 0; i < count; ++i) {
174         iss >> grade;
175         sum += grade;
176         double cumulativeAverage = sum / (i + 1);
177 
178         std::cout << std::setw(3) << std::right << grade
179                   << std::setw(11) << std::right << std::fixed <<
    std::setprecision(2) << cumulativeAverage
180                   << std::endl;
181     }
182 }
183 
184 
```