```
#include <iostream>
#include <iomanip>
#include <string>
#include <algorithm>
#include <limits>
using namespace std;
// Required functions
float calculateAverage(float g1, float g2, float g3, float g4, float g5) {
return (g1 + g2 + g3 + g4 + g5) / 5.0f;
}
float findHighest(float g1, float g2, float g3, float g4, float g5) {
return max(max(max(max(g1, g2), g3), g4), g5);
}
float findLowest(float g1, float g2, float g3, float g4, float g5) {
return min(min(min(min(g1, g2), g3), g4), g5);
}
float getGrade(float avg) {
if (avg >= 96) return 1.0f;
else if (avg >= 94) return 1.25f;
else if (avg >= 92) return 1.5f;
else if (avg >= 90) return 1.75f;
else if (avg >= 88) return 2.0f;
else if (avg >= 86) return 2.25f;
else if (avg >= 84) return 2.5f;
else if (avg >= 82) return 2.75f;
else if (avg >= 80) return 3.0f;
return 5.0f;
}
int countPassing(float g1, float g2, float g3, float g4, float g5) {
return (g1 >= 60) + (g2 >= 60) + (g3 >= 60) + (g4 >= 60) + (g5 >= 60);
}
// Robust input validation (handles non-numeric & out-of-range)
float validatedInput(const string &subject) {
float g;
while (true) {
cout << "Enter " << subject << " grade (0-100): ";
if (!(cin >> g)) {
cin.clear();
cin.ignore(numeric_limits<streamsize>::max(), '\n');
cout << "Invalid input. Please enter a number.\n";
continue;
}
if (g < 0.0f || g > 100.0f) {
cout << "Out of range. Please enter 0 - 100.\n";
```

```
continue;
}
cin.ignore(numeric_limits<streamsize>::max(), '\n'); // clear rest of line
return g;
}
}
bool checkDL(float gwa) { return gwa <= 1.5f; }
int main() {
string name, id;
int age, level;
cout << "Enter name: ";
getline(cin >> ws, name); // ws avoids leftover newline issues
cout << "Enter ID: ";
cin >> id;
cout << "Enter age: ";
cin >> age;
cout << "Enter grade level: ";
cin >> level;
cin.ignore(numeric_limits<streamsize>::max(), '\n');
float math = validatedInput("Math");
float sci = validatedInput("Science");
float eng = validatedInput("English");
float hist = validatedInput("History");
float art = validatedInput("Art");
float avg = calculateAverage(math, sci, eng, hist, art);
float high = findHighest(math, sci, eng, hist, art);
float low = findLowest(math, sci, eng, hist, art);
int passing = countPassing(math, sci, eng, hist, art);
float gwa = getGrade(avg);
cout << "\n===== REPORT CARD =====\n"; //displays the student report card
cout << "Name: " << name << "\nID: " << id << "\nAge: " << age
<< "\nGrade Level: " << level << "\nBirth Year: " << (2025 - age) << "\n\n";
cout << fixed << setprecision(1);
cout << "Math: " << math << "%\nScience: " << sci << "%\nEnglish: " << eng
<< "%\nHistory: " << hist << "%\nArt: " << art << "%\n\n";
cout << "Average: " << avg << "%\nGWA: " << gwa << "\nHighest: " << high
<< "%\nLowest: " << low << "%\nPassing Subjects: " << passing << "/5\n";
cout << "Director's List: " << (checkDL(gwa) ? "YES (Congrats!)" : "NO") << "\n";
return 0;
}
```