Polytechnic University of Puerto Rico
Electrical and Computer Engineering & Computer Science Department
CECS 2203 – Computer Programming I Lab

Lab 2

Name: Coral S. Schmidt Montilla ID#: 148830

1. Copy the source code developed for Lab 2 and paste it as **text** below. (*15 points*)

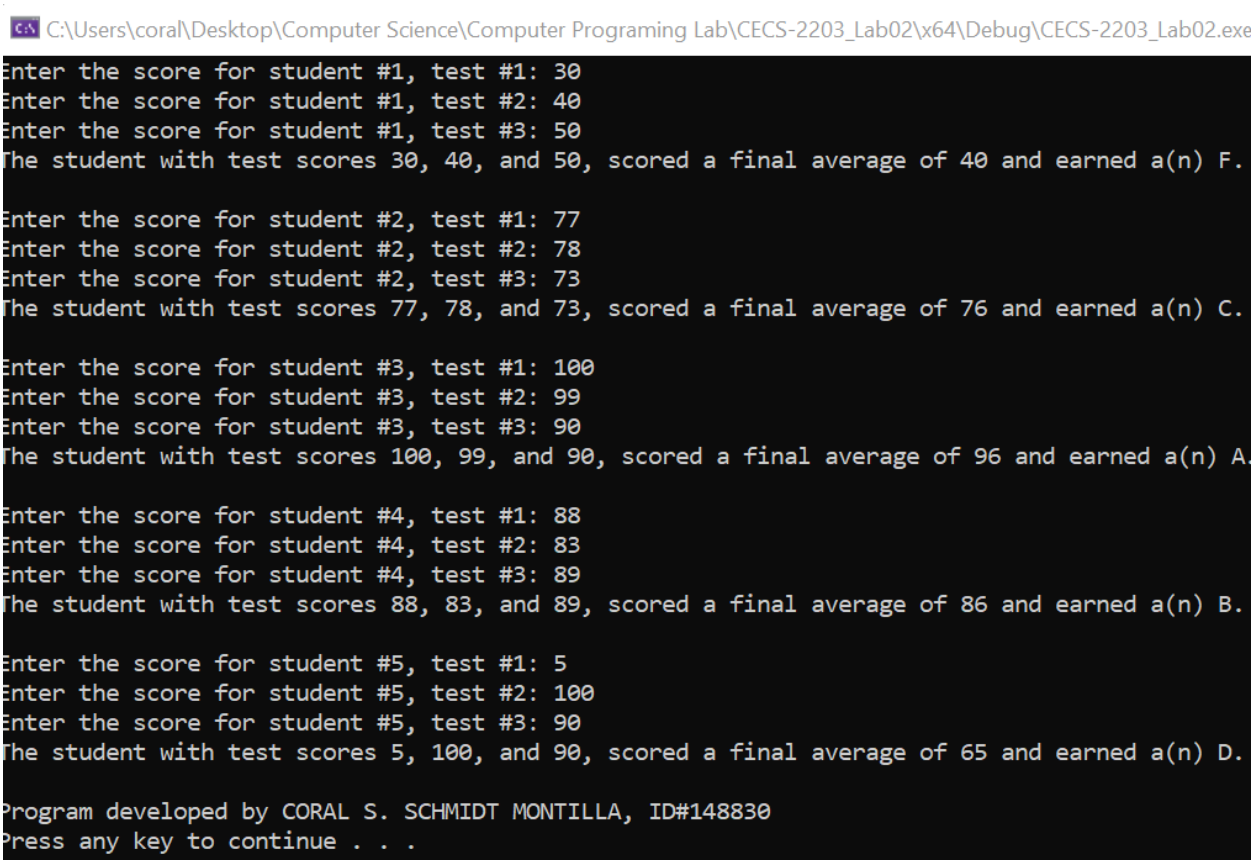
```
/*
* CECS 2203, Computer Programming I Laboratory
* Spring 2023, Sec. 06
* Date: March 23, 2023
* Topic: Lab 2 - Methods that return a value, functions
* File name: lab02.cpp
*
* Instructions and problem statement:
* A college professor needs a solution to compute a student's final letter grade.
* Each student has scores for three 100-point exams. The rubric to be used is:
* -If the final grade is greater than 88, the student earns an A
* -If the final grade is greater than 78, but less than 89, the student earns a B
* -If the final grade is greater than or equal to 69, but less than 79, the student earns a C
* -If the final grade is greater than 58, but less than or equal to 68, the student earns a D
* -If the final grade is less than 59, the student is awarded an F
* The solution must first compute a student's final average, after obtaining all three scores.
* The final average will then be submitted to the rubric, and the final output will be the letter grade.
*
* Complete the program by writing the correct C++ statements.
*
* You should execute the program five times to make sure that all letter grades (A,B,C,D,F) are
* possible. Save a screenshot for each instance of the program's execution, and include them in your report.
*
* Name: CORAL S. SCHMIDT MONTILLA, YOUR ID# 148830
*/
// write the appropriate preprocessor directive
#include <iostream>

// write the appropriate using directive
using namespace std;

// declare a method named getScore that receives 2 integers as parameters
// and returns an integer value
int getScore(int, int);

// declare a method named getAverage that receives 3 integers as parameters
// and returns an integer value
int getAverage(int, int, int);
```

```
40.
41. // declare a method named getGrade that receives an integer as parameter
42. // and returns character value
43. char getGrade(int);
44.
45. int main() {
46.       // Write the statement that declares the integer variables named
47.       // finalAverage, score1, score2, score3 and initializes them to 0
48.       int finalAverage = 0, score1 = 0, score2 = 0, score3 = 0;
49.
50.       // Write the statement that declares a character variable named
51.       // letterGrade and initializes it to X
52.       char letterGrade = 'X';
53.
54.       // Develop a for iteration control structure which uses the integer
   variable i as the counter.
55.       // The structure will repeat the associated block of code 5 times
56.       for (int i = 1; i < 6 ; i++) {
57.
58.             // call getScore to assign a value to score1, use the value of
   the counter as the first argument
59.             score1 = getScore(i, 1);
60.
61.             // call getScore to assign a value to score2, use the value of
   the counter as the first argument
62.             score2 = getScore(i, 2);
63.
64.             // call getScore to assign a value to score3, use the value of
   the counter as the first argument
65.             score3 = getScore(i, 3);
66.
67.             // call getAverage to assign a value to finalAverage
68.             finalAverage = getAverage(score1, score2, score3);
69.
70.             // call getGrade to assign a value to letterGrade
71.             letterGrade = getGrade(finalAverage);
72.
73.             // print the following phrase, substituting the square brackets
   and
74.             // the text within with the correct values:
75.             // "The student with test scores [score1], [score2], and
   [score3],
76.             // scored a final average of [finalAverage] and earned a(n)
   [letterGrade].\n"
77.             cout << "The student with test scores " << score1 <<", " <<
   score2 << ", and " << score3 << ", scored a final average of " <<
   finalAverage << " and earned a(n) " << letterGrade << ".\n";
78.             cout << endl;
79.       }
80.
81.       // write a statement which prints the phrase "Program developed by
   [YOUR NAME], ID#[YOUR ID NUMBER]"
82.       // where the square brackets and the text within is substituted with
   your personal information.
```

```
83.       cout << "Program developed by CORAL S. SCHMIDT MONTILLA, ID#148830";
84.       cout << endl;
85.       system("pause"); // for Visual Studio only
86.       return 0;
87. }
88.
89. // The getScore method receives 2 integer values as parameters and returns an
    integer value.
90. // The value of the first parameter is the student number and the second
    paramter is the test
91. // number, both are used in the prompt. Declare a local integer variable
    named score and initialize
92. // it to 0. Use the following phrase: "Enter the score for student #[student
    number], test #[test number]: ".
93. // Store the value entered by the user in score, and return the value.
94. int getScore(int studNum, int testNum)
95. {
96.       int score = 0;
97.       cout << "Enter the score for student #" << studNum << ", test #" <<
    testNum << ": ";
98.       cin >> score;
99.       return score;
100.      }
101.
102.      // The getAverage method receives 3 integer values as parameters and
    returns an integer value.
103.      // The values of the parameters represent three test scores. The
    function adds the test scores
104.      // and divides the sum by 3, returning this value.
105.      // DO NOT DECLARE any variables in this method!
106.      int getAverage(int score1, int score2, int score3)
107.      {
108.            return ((score1 + score2 + score3) / 3);
109.      }
110.
111.      // The getGrade method receives an integer value as parameter and
    returns a character value.
112.      // The value of the parameter represents the average score for the
    exams. Develop an
113.      // if / else if / else selection control structure to implement the
    rubric, using the value of the
114.      // parameter in the conditions. Use only single conditions! Once a
    condition is true, the function
115.      // returns the corresponding character value, which represents the
    letter grade earned.
116.      // DO NOT DECLARE any variables in this method!
117.      char getGrade(int finalAverage)
118.      {
119.            if (finalAverage > 88)
120.            {
121.                  return 'A';
122.            }
123.            else if (finalAverage > 78 and finalAverage < 89)
124.            {
```

```
125.                    return 'B';
126.             }
127.             else if (finalAverage >= 69 and finalAverage < 79)
128.             {
129.                    return 'C';
130.             }
131.             else if (finalAverage > 58 and finalAverage <= 68)
132.             {
133.                    return 'D';
134.             }
135.             else if (finalAverage < 59)
136.             {
137.                    return 'F';
138.             }
139.             return 0;
140.      }
```

141. Paste the screenshots of the program's execution below. (*5 points*)



142. Comment on any warnings or errors revealed by Visual Studio. If any error messages were present, list the error and describe how you corrected it. If no errors or

warnings were revealed, comment on the most important aspect of developing the solution. (*5 points*)

```
// Develop a for iteration control structure which uses the integer variable i as the counter.
// The structure will repeat the associated block of code 5 times
for (int i = 0; i < 6 ; i++) {

    // call getScore to assign a value to score1, use the value of the counter as the first argument
    score1 = getScore(148830, 1);

    // call getScore to assign a value to score2, use the value of the counter as the first argument
    score2 = getScore(148830, 2);

    // call getScore to assign a value to score3, use the value of the counter as the first argument
    score3 = getScore(148830, 3);
```

```
C:\Users\coral\Desktop\Computer Science\Computer Programing Lab\CECS-2203_Lab02\x64\Debug\CECS-2
Enter the score for student #148830, test #1: 90
Enter the score for student #148830, test #2: 90
Enter the score for student #148830, test #3: 90
The student with test scores 90, 90, and 90, scored a final average of 90 and earned a(n)

Enter the score for student #148830, test #1: 88
Enter the score for student #148830, test #2: 85
Enter the score for student #148830, test #3: 80
The student with test scores 88, 85, and 80, scored a final average of 84 and earned a(n)

Enter the score for student #148830, test #1: 77
Enter the score for student #148830, test #2: 71
Enter the score for student #148830, test #3: 69
The student with test scores 77, 71, and 69, scored a final average of 72 and earned a(n)

Enter the score for student #148830, test #1: 55
Enter the score for student #148830, test #2: 51
Enter the score for student #148830, test #3: 59
The student with test scores 55, 51, and 59, scored a final average of 55 and earned a(n)

Enter the score for student #148830, test #1: 67
Enter the score for student #148830, test #2: 90
Enter the score for student #148830, test #3: 0
The student with test scores 67, 90, and 0, scored a final average of 52 and earned a(n) F

Enter the score for student #148830, test #1: 0
Enter the score for student #148830, test #2: 0
Enter the score for student #148830, test #3: 0
The student with test scores 0, 0, and 0, scored a final average of 0 and earned a(n) F.

Program developed by CORAL S. SCHMIDT MONTILLA, ID#148830
Press any key to continue . . .
```

The error was that it was printing 6 times instead of 5. I realized that I had to initialize the i = 1 and I also had to replace student with i instead of placing the number.