CIS 246 – C++ Programming

Program:	**Week 05**
Points:	**20**
Chapter(s):	**6**
File(s) to Submit:	**OverloadDriver.cpp, Average.h (as one zip file)**

Summary

Write a C++ application that uses overloaded functions to determine the average of numbers entered by the user.

Description

Write a driver program that prompts the user per the instructions below. Call one of four overloaded computeAverage functions located in the Average class. Display the results of each function call.

You must write **a driver and a class** using separate files for full credit.

NOTE – Use of any math libraries is not allowed for this assignment.

Learning Objectives

In this assignment, you will practice:

- Writing overloaded functions
- Using overloaded functions
- Creating a driver program and a class

Requirements for the Average Class

Data Members

None

Member Functions

Write four public computeAverage functions. Each version will use a different parameter list:

1. 2 integer parameters
2. 3 integer parameters
3. 2 double parameters
4. 3 double parameters

All four functions will use the same return type – always a double.

Within each function, compute and return the average of the values passed in. You may not use any math libraries for this. Write your own functionality to determine the average.

Requirements for the OverloadDriver Program

All of the logic in OverloadDriver.cpp will be in a main function.

Prompt the user four times:

1. Enter 2 integers, separated by a space, on a single line:
2. Enter 3 integers, separated by spaces, on a single line:
3. Enter 2 doubles, separated by a space, on a single line:
4. Enter 3 doubles, separated by spaces, on a single line:

The user must type on the same line as the prompt.

After each prompt:

1. Read the two numbers typed in.
2. Call the appropriate overloaded computeAverage function from the Average class. You do not need to use selection for this step!
3. Display the result of the function call. Be sure to use the appropriate precision if necessary.

You do not have to write a loop to prompt for data or to read what the user types.

Sample Run

```
Enter 2 integers, separated by a space, on a single line: 10 20
Average of 10 and 20 is 15.00

Enter 3 integers, separated by a space, on a single line: 10 20 30
Average of 10 and 20 and 30 is 20.00

Enter 2 doubles, separated by a space, on a single line: 10.0 20.0
Average of 10.00 and 20.00 is 15.00

Enter 3 doubles, separated by a space, on a single line: 10.0 20.0 30.0
Average of 10.0 and 20.0 and 30.0 is 20.00
```

Requirements for Full Credit on This Project

SUBMIT YOUR OWN WORK – Plagiarism is not tolerated in this course. Please review the section on the Academic Honor Code in the syllabus. I will not hesitate to drop you from this class if you submit a program that is plagiarized.

COMPLETE AND ACCURATE – Your program must compile, execute, and give accurate output.

FOLLOW ALL REQUIREMENTS ACCORDING TO THE INSTRUCTIONS – Follow the instructions as written for completing this project, even if you [think you] know a "better" way to do something.

COMMENTS – Include comments in your code. There must be a comment at the top of each source code or header file that includes your name, the assignment number, and a description of the code in that file. There must be comments at each important step in your algorithm that describes that step.

FOLLOW BEST PRACTICES – Follow best practices in C++ programming, including, but not limited to, appropriate use of private/public, appropriate use of classes and/or header files, sets & gets, white space, alignment, meaningful variable names, naming conventions, using statements, etc. Points will be deducted for sloppy code that is hard to read, even if it works, so pay attention to these details.

SUBMIT ALL FILES BEFORE THE DUE DATE – Submit a .zip of ONLY source code files to the dropbox for this assignment on Canvas before the due date. Do not submit anything except .cpp and/or .h, within a zip. Do not submit .exe files. Do not submit a folder structure. Do not submit project files from an IDE.