CPSC 240 Assignment 3

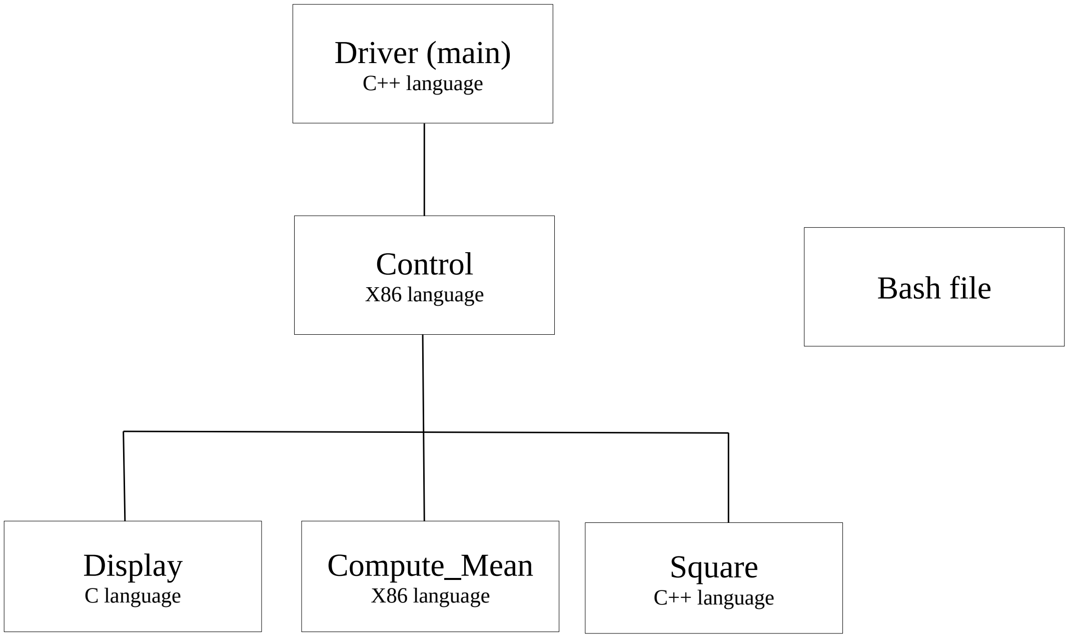
Arrays

Educational objectives

The educational objective of this assignment is that you personally have the experience of do all of the following. Reading about these activities in a book or asking someone else to do this program for you does not reach the objective.

Declare an array
Pass data to a subprogram
Convert an integer to a numerically equal floating point number
Pass a float number to another function
Use a GPR register in the role of index to an array.

Program structure



Five source files

The module called “driver” does not know anything about what this program does. Its principle job is to call the control module. When the control module has terminated this driver has received a number having no special significance to the driver. So, the driver simply prints that number.

The “control” module is really in charge of this program. It declares the array. It asks the user to input data into the array. It calls submodules: display, compute_mean, and square. At the end it returns the computed mean.

The module “display” performs only one task: output in an orderly fashion the data in any array passed to it.   This module does not display any welcome messages nor any general information.

The module compute_mean does what its name says. First it finds the sum of all the numbers in the array. Then it divides the sum by the count of how many numbers are present. The quotient is now called mean. The mean is returned to the calling program. This module does not output anything.

The module “square” changes every value in the array to become the square of the value. This module does not output anything by itself

The bash file runs the entire show. It compiles 5 sources files and then links them into a single executable file, and then executes that file. You already know how to do this.

Sample session while running this program.

```
Welcome to an array of long integers.
This program is brought to you by Mai Hong Lam.

The control module has begun.
Instructions: Enter a sequence of integers.  Include white space between each number.
To terminate press ‘Enter’ followed by Control+D.
73   56   95    27    82     9  <enter><cntl+D>
Here are the data as received:
          73
          56
          95
          27
          83
           9
The mean of these 6 numbers is 57.1667
Here are the squares of the data:
          5329
          3136
          9025
           729
          6889
            81
The control module is now returning to the caller module.  Bye.

The driver received this unknown number:   57.1667
The driver will now return 0 to the operating system.
```

Color codes: Text in red is output from the main function (C++)

Text in black is output from the controller (x86)

Text in blue is output from the display module (C)

Professionalism

This is one program made from five different files (modules). This one program needs a title. You have to imagine a nice title. Maybe something like "How to Average Integers" or "Practical Use of an Array". You think of a title. Then that one title appears in the comments of all 5 source files.

The following text appears in all five source files of this program.

Author name: xxxxxxxxx xxxxxx
Program name: How to Average Integerss
Names of files in this programming: driver.cpp, control.asm, display.c, mean.asm, square.cpp
Course number: CPSC 240
Scheduled delivery date: March 7, 2019
Program purpose: Software to assist teachers in computing a class average.
Status: Finished. No known bugs
Date of last modification: Feb 28, 2019
Information about this module:
This module purpose: Driver for How to Average Integers
File name of this module: integers.cpp
Compile this module: g++ -c -m64 -std≡c++99 -o integers.o integers.cpp
Link this module with other objects:
g++ -m64 -std=c++99 -o average.out integers.o controller.o display.o mean.o display.o

Put the green text or something equivalent to it in all 5 of the sources files. It would be a good idea to include it in the bash file as well. What is the name of the folder where all 6 files will be found? That's right: in the folder How_to_average_integers.

Do not leave the professor's name in any source files for which you intend to receive credit.

Portfolio

It is never too late to start making your own portfolio. In each computer class you take save your documents related to the assignment where you received your best grade that semester. The set of documents will go into a Professional Portfolio.

Suppose there is a programming assignment in CPSC 332, where you did an outstandingly good job. Save documents such as the assignment, your source files, output from a few sample executions, and any thing related to that one programming project. Create a document to add to the end of the collection with a title such as “Lessons Learned from Database Project”. In the “Lessons Learned” document record what you really learned in doing that project. The whole set of documents becomes a section in your portfolio.

Make one section for almost every CS course you complete. Keep all the sections in a 3-ring binder. Create a nice title page for each section. Create a title page for the whole porfolio.

Take the portfolio with you on job interviews. Briefly state that you have available with you a portfolio of projects you did.   Some interviewers will gladly peruse your portfolio and others will show no interest. But, you did your part, namely: you made it available.

Some few people says to simply post your best projects on github and provide the prospective employer with a link. You may decide to do it that way if you wish. However, if I were the employer I would be more impressed by someone who came to me with organized professional-looking project material. You decide which is best suited for you.

Testing and verification

Test your own program. It is psychologically difficult to break your own program, but try anyway. Use some negative inputs. Use no inputs. Use exactly 1 input. Use too many so that now all will fit in the array. Those are the kinds of tests the professor will do with your program if you send it.

Due date: March 7, 2019.

If you deliver a completed assignment #3 to me on or before the due date then you are fully up to date with the programming of this course. Similarly, if you complete Assignment 3 after the given date you are falling behind the normal progress of the course.

Mail to: **holliday@fullerton.edu**

Footnote: If you open this file with Microsoft Word any diagrams (like the one on page 1) will be distorted. You should open this file with Libre Office Writer, which is the same program that created this file. You will find Libre Office Writer in Tuffix. You can also download a version of Libre Office Writer for Windows.