# Miami University
# College of Engineering and Computing
# Department of Computer Science and Software Engineering

CSE-278 Systems I  Fall 2018

Lab 04

*In this lab, we are going to continue practice NetBeans, this time with a program involving files and command line arguments. We also learn about a useful GNU-Linux utility to compare files.*

You will find the source and data files in

`/home/cse-278/WK04`

This time it has the files

```
lb04.pdf
ex-lab04.cpp
exercise4.cpp
ex4_expected_output.html
ex4_inputs.txt
exercise4.cpp

FinancialAidAward.cpp
FinancialAidAward.h
StudentAidRecord.cpp
StudentAidRecord.h
testdriver.cpp
```

The first one is the **.pdf* file that you are reading.

| Work through the worksheet `ex-lab04.docx`, and submit it to grading by the due date. Then work on the supplementary problems. |
|---|

This supplementary exercises are about simple progamming in C++. Please feel to write this programs in any way you like it. If you use an IDE, make sure you can compile and run by command line.

Bear in mind these are not graded, but you should work on them. Create a directory for each week and put your work in there. It will be invaluable to review for the midterms!.

1. The function $H(n)$ is defined

$$H(n) = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + ...\frac{1}{n}.$$

   Write a program that includes a function that returns $H(n)$. Using a loop from the main program compute $H(10), H(100), H(1000)$ Show your results with at least 6 decimal places (using `iomanip` library) Make sure you use the best primitive data type for this type of problems.

2. Compile, link and execute the files at least 3 different ways, using g++ (with/without the -o ). You may experiment with the linking loader `ld`.

3. Study the programs:

   `FinancialAidAward.cpp`
   `StudentAidRecord.cpp`
   `testdriver.cpp`

   Without running, determine what the program does. Pay particular attention to the use of array of objects, and how I/O is coded, and the use of `typedef`.

4. Compile, link and execute the files at least 3 different ways, using g++ (with/without the -o ). You may experiment with the linking loader `ld`.