# COMS W4115: Programming Assignment 3

Write a simple LLVM pass

## Logistics

1. **Announcement Date:** October 7th, 2019
2. **Due Date:** October 16th, 2019 by 5:00pm. No extension!!
3. **Total Points:** 100

## Write a simple LLVM pass

1. Run an example LLVM pass

   (a) Run the following command to compile bubble.c to an optimized LLVM bytecode object bubble.bc.

   ```
   ./llvm-project/build/bin/clang -O -emit-llvm -c bubble.c
   ```

   (b) To inspect the generated bytecode bubble.bc, run the following command on bubble.bc to generate bubble.ll.

   ```
   ./llvm-project/build/bin/llvm-dis bubble.bc
   ```

   (c) Run the following command to first make a copy of bubble.bc and bubble.ll, then to compile bubble.c to an LLVM bytecode object without optimization.

   ```
   cp bubble.bc bubble_with_optimization.bc
   cp bubble.ll bubble_with_optimization.ll
   ./llvm-project/build/bin/clang -O0 -emit-llvm -c bubble.c
   ./llvm-project/build/bin/llvm-dis bubble.bc
   ```

   (d) Compare the new bubble.ll with the previously generated bubble_with_optimization.ll and write a paragraph about what difference you observe in **{UNI}.txt** and submit it.

   (e) Use the following command to run an example LLVM pass on bubble.bc.

   ```
   ./build/bin/opt -load ./build/lib/LLVMHello.so -hello < bubble.bc > /dev/null
   ```

   (f) Read the source file in folder "llvm-project/llvm/lib/Transforms/Hello" and read the following website for detail explanation.

   http://llvm.org/docs/WritingAnLLVMPass.html#writing-an-llvm-pass-basiccode.

2. Create an LLVM pass

   The example Hello pass prints each defined function name. In this assignment, you are required to create another LLVM pass(don't modify the hello pass) to print the following information for each defined function.

```
(1) Name
(2) Number of Arguments (* if applicable)
(3) Number of direct call sites in the same LLVM module (i.e.
locations where this function is explicitly called, ignoring
function pointers).
(4) Number of Basic Blocks
(5) Number of Instructions
```

In order to create a new LLVM pass, firstly, you may need to create another folder in folder "llvm-project/llvm/lib/Transforms/". Create files in the new created folder according the files in "Hello" folder. Then edit CMakeLists.txt in both "Transforms" folder and the new created folder. Finally run "make" under "llvm-project/build" folder to build the LLVM pass.

Note: LLVM documentation will be helpful. https://llvm.org/doxygen/classllvm_1_1Function.html

Please rename your cpp file to {UNI}.cpp and submit it. During grading, we will first rename your cpp file to hw3.cpp and run it as follows.

```
./build/bin/opt -load ./build/lib/LLVMhw3.so -hw3 < bubble.bc > /dev/null
```

Please define the created class name and registered LLVM pass name accordingly for full grade. using "swap" as an example, if bubble.bc is optimized bytecode, it should output:

```
swap arguments=2 callsites=0 basicblocks=1 instructions=5
```

If bubble.bc is the bytecode without optimization, it should output:

```
swap arguments=2 callsites=1 basicblocks=1 instructions=16
```

We may also test your program on other C programs besides bubble.c.

## Submission Guide

Please submit the followings:

1. You are required to submit **{UNI}.txt**(20 points) and **{UNI}.cpp**(80 points). {UNI} means your UNI number.
2. For all assignments, submit an extra file **contribution.txt** describing each of your contribution.