

# CS6380: Assignment #3

## Loop Analysis Pass

Due Wednesday September 8<sup>th</sup>, 2016 at 11:59 PM

The aim of this assignment is to familiarize you with writing LLVM analysis passes. You will write a simple pass to analyze loops.

### 1 Loop properties

Your pass should collect the following information for all the loops in the program:

- Function name. [string]
- Loop depth. [int]
- Does it contain any loops? [bool]
- No. of top level basic blocks: basic blocks in it but not in any of the nested loops. [int]
- No. of instructions: this should include the instructions in the nested loops. [int]
- No. of top-level branch instructions: branch instructions in it but not in any of its nested loops. [int]

For each loop print a line to standard error in the following format:

```
<ID>: func=<str>, depth=<int>, subLoops=<bool>, BBs=<int>, instrs=<int>, branches=<int>
```

<ID> represents a **global** counter for each loop encountered, starting at 0.

### 2 Implementation Guidelines

Create a new directory LoopProps in `lib/Transforms/` folder of the LLVM source tree. All your code should be in this directory as this directory is part of your submission. Register your pass as `loop-props` so that we can run the pass using `opt -load $LLVM_BUILD/lib/LoopProps.so -loop-props input.ll`

### 3 Testing

You are supposed to write test cases to test your pass. They should vary in complexity from simple to more complex ones. You need to submit 5 test cases.

## 4 Submission

Your submission should be an archive containing the LoopProps directory, a test-cases directory and a Readme file. The Readme should mention all the materials that you have read/used for this assignment including LLVM documentation and source files. Mention the status of your submission, in case some part of it is incomplete. You can also include feedback, like what was challenging and what was trivial.